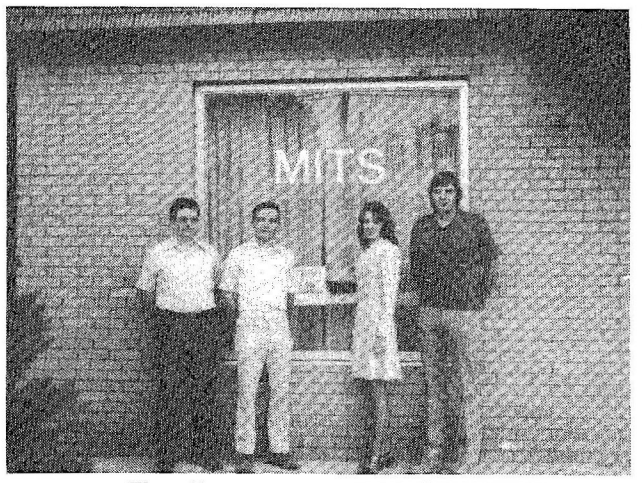
BYTE'TRONICS TO SELL ALTAIRS

BY Lloyd J. Austin

The telephone rings in a small but comfortable store-front office in west Knoxville, Tennessee. A moment later a pleasant voice answers, "MITS - Byte 'Tronics; how can we help you?" Sandra Seals, who is most likely to be the first voice you meet when you call, makes you feel at ease and yet gives you the assurance that you will get prompt and courteous service, no matter what your reason for calling. This is natural with her because she knows that she is working with well-trained and competent people who solve computer problems every day.

Whether you have called for service, for information about the newest of the line of Altair Microcomputers, for a suggestion in a software program, or any other of the myriad of questions that just come naturally, you will get your answers quickly and efficiently. If you have a question concerning the availability of a MITS product, its price, its specifications, or other similar information, Sandra will switch you over to Bruce Seals, who is the marketing director and



The People at BYTE'TRONICS
(l to r) Hugh, Bruce, Sandra, Johnny

sales manager. If you need to know about the compatibility between your Altair 8800 and some input or output device, you will probably talk with Technical Director Hugh Huddelston about it. When you need a software program with special modifications and subroutines, Johnny Reed is your man, and he is familiar with programming on all of the several languages that can be used. All personnel are well-qualified to help you with any service problem because they come from a background of computer applications and are thoroughly familiar with the MITS circuits in Altair, including the optional boards and peripheral devices that can be used.

The office of Byte 'Tronics is located at 5604 Kingston Pike, Knoxville, Tennessee - Zip Code 37919. Everyone is welcome to drop in at any time, to call on the phone at (615) 588-8971, or to make contact by letter. The office hours are (Eastern Time Zone) 10AM to 10PM Monday through Friday and 9AM to 10PM on Saturday. That should allow everyone who is interested to find the office open at some convenient time.

Operating as an independent distributor, Byte'Tronics sells the full line of MITS products. The Altair and its peripherals are available in either kit or ready-to-use form and, whenever possible, on an off-the-shelf timetable. The prices are the same as those in effect at the factory, including special sale prices when they are offered. But the sale and delivery of the equipment is only the beginning of a long and fruitful relationship between every customer and Byte 'Tronics, because their service--both hardware and software—is part of the package that you get when you trade with them. Byte 'Tronics is equally at home with the hobbyist or the professional engineer, the university student or the production line foreman, or any of the many other people who are finding new uses every day for computers-now that they are available and affordable. Hugh and Bruce and Johnny form an excellent team to furnish this client-supplier relationship because they have all been involved in every aspect of the business as users, fixers, testers, and operators.

Hugh's pride and joy is the fully stocked parts cabinet in the back room, with the electronic equipment that will help chase down the trickiest failure. His thorough knowledge of the purpose for each portion of the circuit on the board, together with his experience in troubleshooting, makes him very effi-

Continued on page 3---

COMPUTER NOTES

September, 1975

Volume One Issue Four

A PUBLICATION OF THE ALTAIR USERS GROUP

MITS-MOBILE TOURS NE

Nearing the completion of a highly successful tour of the Southeastern U.S., the MITS MOBILE "Van Man," Mike Hunter, reports SRO crowds in most cities and an enthusiastic response from all those participating. The Northeast and Midwest are up next, with the first stop of the fall tour set for Buffalo, NY on October 20. (for complete schedule, see page 3.)

When the MITS MOBILE comes to your area, you will have a unique opportunity to attend a seminar and slide presentation covering a wide range of topics including computer concepts, technical aspects of computers in general and computer programming. This is your chance to hear some straight talk about computers and to ask any questions you may have--from the most "simple" to the most "advanced." Mike wants to show you how accessible low-cost computing really is.

All the seminars are held at Holiday Inns. Hands-on demonstrations of a complete Altair Basic Language System begin at 6pm, the seminar itself runs from 7-10pm, and after 10pm more discussions or demonstrations for anyone who wishes to stay.

Everyone present will receive a three-ring binder loaded with course material, Altair data and schematics, and catalogs. Reservations cost $12 and must be made in advance by mailing your check (BankAmericard or MasterCharge accepted also) to MITS in Albuquerque. You will receive your admission ticket and the exact location of the seminar by return mail.

MITS/6328 Linn NE/Albuquerque, NM 87108/ 505-265-7553

OR 505-262-1951

********** BULLETIN **********

We have discovered that many Signetic 2604 4K RAM's found on some of our 4K Dynamic Memory Boards do not meet the required specifications for access time and refresh period. They are identified on the package as S 2604. If you have an Altair 4K Board that does not work properly and it has Signetic RAM's, please return the board along with the Signetic RAM's for complete checkout with replacement RAM's. We are sorry for the inconvenience.

Across the Editor's Desk

by David Bunnell

RANCE CRAIN of AD AGE

Rance Crain, the editorial director of Advertising Age and Crain Communications, was in Albuquerque recently to address the local ad club.

Now this isn't the type of news item that would normally be of interest to Altair users, but Mr. Crain said some very interesting things. For example, it is his contention that business has been so stifled by government regulations and consumer protection groups that managers are no longer able to make innovative, creative decisions. And the end result of this is a lack of imaginative products.

My initial reaction is that Mr. Crain is unaware of the electronics business. However, in arguing his position, Crain made some very convincing points about the changing values of consumers and the slowness of catch-up by industry. This is particularly evident in the automotive industry where Detroit has been slow to see the growing consumer demand for cars that are efficient, reliable and safe.

According to Crain, the consumer code of the 60's was "Buy, Buy, Buy." This is no longer in effect. Consumers are more cautious now and, most importantly, consumers are smarter.

The MITS Philosophy

MITS is fortunate that it is a relatively new corporation. Its philosophy is in the formulative stage where it still can be influenced by changing consumer attitudes and existing economic realities.

For one thing, MITS is a very efficient corporation in that money isn't wasted for frills. We make sure that our products get the best engineering expertise but carpeting and pretty packaging are low priorities.

MITS tries not to commit itself to positions that it can't keep, and we try to level with our customers when something goes wrong. Still, we are dictated to by the rules of reality and sometimes someone pokes us in the eye. And sometimes we deserve it.

Consumer Responsibility

The business ethics of the 70's has so far failed to focus on the responsibilities of the consumer. If business is to adhere to new rules, if it is to eliminate misleading advertising, be conscious of the environmental impact of its products, avoid making excessive profits, etc., what then, if anything, should the consumer do in return?

Now, I don't pretend to have the answer to this question but morality is a two way street, and one thing that bothers me is the fact that a few of our customers have been ripping off MITS software. In violation of their software license some people have been arrogantly, and I think foolishly, copying MITS BASIC for resale or to pass out to their freinds.

I believe this practice is fostered by the contention that all software should be made part of the public domain.

Now I ask you--does a musician have the right to collect the royalty on the sale of his records or does a writer have the right to collect the royalty on the sale of his books? Are people who copy software any different than those who copy records and books?

Altair BASIC is one of the most advanced BASICs ever written and it cost MITS a premium price to develop. Considering this and the price charged by many companies for software, $75 for Altair 8K BASIC is a near steal. And the price of $500 for people who haven't purchased a minimum 8800 system is more than reasonable.

Since I've spouted off, I want to invite Altair customers to spout off to us if they've encountered any difficulties such as poor service, delivery, etc. Call us at 505-262-1951 or address your letters to the Altair Customer Service Department.

We can't go about changing the whole MITS mode of operation everytime someone sends us a complaint--but, believe me, your letters will be read and seriously considered by someone in a high management position.

Final Message

Five or six years ago MITS was literally a garage operation not unlike many of the new computer hobby companies that have recently sprung up here and there. MITS was successful because it was able to deliver what it advertised and the success of these new companies will likewise be highly dependent on their credibility.

Most of the people in this business have high ethical standards but watch out for the few ripoff artists. Don't order anything until you know it is "real."

ALTAIR SERVICE DEPT.

Barbara Sims

Hello Again!

We now have several programs in our Users Group library. Descriptions of these have been published in our past and present newsletters. We are also printing prices of the programs in Computer Notes regularly. If you have a program that would be of general interest, please send it in.

It has been a great help now that programs are being submitted on the program submission forms, however there has been a slight mix-up concerning the form. The coding form we sent you in the August issue of our newsletter was a sample. The sample should be used to make up an original and have it printed if you wish. Then, use the printed copies to send in to us. We can print directly from your program if it is handled in this way. However, if you merely photocopy or xerox our sample, we have to retype the entire program. This slows up printing and in the end slows up delivery of the programs to you. The same is true of a typewriter with an old ribbon or dirty keys. If the type is not clear and of good contrast, then we have to retype the program for printing. This all may sound very particular, but we are trying to cut down the handling time so that programs sent in will immediately be available to you, the user.

If you'd like a simpler course to follow, you can order the program submission forms directly from MITS, 50 copies costing $2.00. This is a fairly reasonable price compared to local printing companies. At any rate, keep the programs coming in.

Our Marketing Department needs some help from customers also. Anytime you call in an order to MITS of any type, it is critical that you give the name your 8800 was ordered under and if at all possible, the 8800 invoice number. This helps us in our record keeping between our accounting department, marketing department, and service department.

Adios!

Barbara

Byte 'Tronics continued---

cient. If you need an interface to drive some particular piece of gear, Hugh is most likely the one to make it - or at least to supervise how it is made.

Johnny has recently been spending some of his time at several customers' places of business. He has been writing complete software programs to custom requirements. He is a good man to know when you need help. of course, he can operate any system, simple to complex, and he can spot an error in your program about as fast as the Altair can signal "Error."

Johnny and Bruce are especially interested in a new and growing function - the local Users Group of East Tennessee. All owners of an Altair 8800 are eleigible for a year's free membership in this group, in keeping with the MITS plans. This group meets once a week to swap information and just "talk computers." Sharing ideas and programs is their purpose, and this should be helpful to all whether they are novice or experienced. If you would like more information, drop a post card to them at Byte 'Tronics and you will get the same quick and courteous response as if you asked to buy $10,000 worth of equipment!

So if you are among those who are fortunate enough to already be an Altair user, or if you just want to be, either way you should get acquainted with your friends at Byte 'Tronics because they are well qualified and anxious to help you. All you need is a post card, a letter, a phone call, or - if it is convenient - a visit to the office in Knoxville, and your world will open up in front of you.

MITS SALES REPRESENTATIVES

For detailed information on Altair computers and computer products, contact the MITS sales representative in your area. Or call our factory direct:

MITS/6328 Linn NE
Albuquerque, NM 87108
505-265-7553 or 262-1952

MIDWEST

Inland Associates
13100 Manchester Road #G-25
St. Louis, MO 63131
314-821-3742

Inland Associates
2310 West 75th
Shawnee Mission, KS 66208
913-362-2366

Ridgway East, Inc.
15326 Garfield Road
Detroit, MI 48239
313-538-3940

Ridgway East, Inc.
138 Elseden Building
Florence, KY 41042
606-371-1269

Ridgway East, Inc.
173 Hawthorn Drive
Chagrin Falls, OH 44022
216-247-4845

SOUTHEAST

MITS, Inc.
Southeastern Regional Office
5508 NW 72nd Avenue
Miami, FL 33166
305-885-9388

COL-INS-CO, Inc.
1313 44th Street
Orlando, FL 32809
305-423-7615

COL-INS-CO, Inc.
Offices in:
Huntsville, AL
Largo, FL
Atlanta, GA
Raleigh, NC
Call 800-327-6600 Toll-free

WEST COAST

REPCO
50 East Middlefield Road
Mountain View, CA 94043
415-965-8581

CTI Data Systems
3450 East Spring Street
Long Beach, CA 90806
213-426-7375

CTI Data Systems
8869 Balboa Avenue C
San Diego, CA 92123
714-292-0636

REPCO
PO Box 811
Bellevue, WA 98004
206-455-1246

WESTERN U.S.

BFA Corporation
1350 Chambers Road #207
Aurora, CO 80011
303-344-3800

BFA Corporation
395 Lawndale Drive
Salt Lake City, UT 84115
801-466-6522

BFA Corporation
4251 North Brown Avenue
Scottsdale, AZ 85251
602-946-4215

BFA Corporation
9004 Menaul Boulevard NE
Albuquerque, NM 87112
505-292-1212

BFA Corporation
Dona Ana Road, PO Box 1237
Las Cruces, NM 88001
505-523-0601

NORTHEAST

J. J. Wild, Inc. of New England
PO Box 382
Needham, MA 02192
617-444-2366

J. J. Wild, Inc. of New England
PO Box 342
Southbury, CT 06488
203-264-9494

J. J. Wild, Inc.
73 Main Street
Woodbridge, NJ 07095
201-636-7780

J. J. Wild, Inc.
80 Second Street Pike
Southampton, PA 18966
215-357-6645

J. J. Wild, Inc.
400 Jericho Turnpike
Jericho, NY 11753
516-935-6600
(for Western New York and Western Pennsylvania, see:
Ridgway East, Inc.
Chagrin Falls, OH)

NOTE: Sales Representatives are geared to serving industrial customers. Hobbyists should contact the factory directly or one of the MITS distributors. Distributors now include Byte'Tronics in Knoxville, Tennessee (see page 1 article) and The Computer Store in West Los Angeles (11656 Pico, phone 213-478-3168). Other MITS distributors will soon be set up across the country. Watch Computer Notes for all the details.

MITS-MOBILE CARAVAN SEMINAR SCHEDULE

GROUP A -- Reservation Deadline October 10, 1975

October 20	Buffalo, NY	October 30	Hartford, CT
October 21	Rochester, NY	October 31	White Plains, NY
October 22	Syracuse, NY	November 3	New York City
October 24	Albany, NY	November 4	Hempstead, LI, NY
October 27	Boston, MA (Cambridge)	November 6	Plainview, LI, NY
October 28	Boston, MA (Newton)	November 7	Hackensack, NJ

GROUP B -- Reservation Deadline October 25, 1975

November 10	Allentown, PA	December 1	Chicago (Northbrook), IL
November 11	Philadelphia, PA	December 2	Milwaukee, WI
November 13	Baltimore, MD	December 4	Minneapolis, MN
November 14	Washington, DC	December 5	Madison, WI
November 17	Pittsburgh, PA	December 8	Indianapolis, IN
November 19	Columbus, OH	December 9	Cincinnati, OH
November 20	Cleveland, OH	December 10	Louisville, KY
November 21	Detroit, MI	December 12	St. Louis, MO
November 24	Kalamazoo, MI	December 15	Davenport, IA
November 25	Chicago (Hinesdale), IL	December 16	Des Moines, IA
		December 17	Omaha, NE
		December 19	Kansas City, MO
		December 21	Wichita, KS

88-VLCT MOD

88-VLCT READY KEY MOD

PROBLEM:
Pressing "READY" key should cause one strobe pulse to PIO board "SBO" line, causing computer to output data to octal display. Noise from keyswitch bounce causes multiple pulses on "SBO" line, causing next byte entered to be echoed without pressing READY.

SOLUTION:
Change R32 from 10K to 10meg. R32 is across C6, the .01 pulse timing capacitor for the READY key. Increasing R32 to 10meg makes discharge time for C6 greater than 10ms, preventing keyswitch bounce.

NOTE:
READY key schematic is incorrect: R33, 100Ω, shown going to Vcc is actually connected to ground. R31, 47Ω, shown going to ground, actually goes to Vcc.

10meg -- MITS part number 102079

USERS' NAMES & ADDRESSES

Frank E. Corlett
205 Port-O-Call
Bridgeport, TX 76026

Keith L. Kendall
295 E. 500 S.
Vernal, UT 84078

Glenn Nelson
160 Greenway W.
Newhyde Park, NY 11040

Mount St. Mary's
c/o Bill O'Toole
Emmitsburg, MD 21727

Sgt. Wesley B. Isgregg
Box 3558 C & E Schools MCB
29 Palms, CA 92276
714-368-9111 DWH Ext. 6241
AWH Ext. 7289

Sgt. Stanley E. Herr
13-C Copper Dr. MCB
29 Palms, CA 92278
714-368-3809

Robert Beard
2530 Hillegass Apt. 109
Berkeley, CA 94704

Tod Rapp
129 Park Dr.
Xenia, OH 45385
Business 513-372-8294
Home 513-372-8445

Jim Fisk-WB9N1V
4116 Euclid Ave.
Ft. Wayne, IN 46806
219-745-0359

Michael A. Enkelis
9924 S.W. 31st Ave.
Portland, OR 97219
503-246-4614

Stephen E. Clark
2099 Powder Horn Dr.
Clearwater, FL 33528

Bruce Segal
64 Summit Crescent
Westmount PQ
Canada H3Y 1L6

C. A. Kirkpatrick
2041 San Sebastian Ct.
Apt. 70
Houston, TX 77058

Richard Hole-WA8TSY
Gerber Products Company
Fremont, MI 49412
Business 616-928-2692
Home 616-652-6884

Gary Tack
P.O. Box 866
Corrales, NM 87048

Matthew W. Smith
4355 S. High St.
Englewood, CO 80110

Gary S. Trent
Route 5, Box 900
Space 17
Orange, TX 77630

John E. Kabat
"Universe Unlimited"
User's Group
11918 Forrest Ave.
Cleveland, OH 44120
Business 216-781-9400 Ext.55
Home 216-795-2565

D. Minott
352 Arkansas Dr.
Valley Stream, NY 11580

Donald C. Schertz
764 Toyon Dr.
Monterey, CA 93940

Ivan Wampfler
7861 Old River Rd.
Rockford, IL 61103
815-633-4757

Richard Bushick-WA3PWO
418 Brian Ct.
Mechanicsburg, PA 17055

John Rabenaldt
1301 N. Jackson #24
Odessa, TX 79761
915-337-6931

To be featured on cover of November, 1975 Popular Electronics.

SNEAK PREVIEW

In January of 1975, MITS stunned the computer world with the announcement of the Altair 8800 general purpose computer that sells for $439 in kit form and $621 assembled.

In October of 1975, MITS will announce a complete computer built around the 6800 MPU available from Motorola and AMI.

This computer will come with an MPU board that has 1K RAM, built-in I/O that can be configured three ways, and provision for 1K ROM or PROM. It will have power supply and be sold with front panel control board in an 11" x 11" x 4 11/16" case for $293 in kit form and $420 assembled.

The MPU Board—ideal for dedicated control applications—will be marketed for $180 in kit form and $275 assembled.

For complete details, see November's Popular Electronics.

Aug Software Contest Winners Announced

by Paul Allen and Bill Gates

This month nine programs and nine subroutines were added to the Library.

The ultimate in memory clears seems to have been written by Ward Christensen (#731751).

"I have been using the following 'program' to clear memory in my Altair since the first week I had it. It clears all of memory except byte 0, and leaves the address in location 1. It works because the Altair seems to have the stack pointer at location 0 when powered up. At worst, it would have to be run twice."

```
0000   063   INX SP
0001   307   RST 0
```

The winning major program this month is a quadruple byte integer manipulation package by Steve Phillips (see below). In second place is R. J. Walker's PIO BASIC or Package I loader. Third place goes to John Trautschold for his mutiplication program for floating point numbers.

In the subroutine category, first place goes to Donald Tork for his table search routine. Second place goes to Jonathan Griffitts for a subroutine which may be used to display the A register in the status lights for one second, which could be useful as a debugging tool.

PROGRAMS

#92751
Author: Dean B. McDaniel
Length: 23 bytes
"Object: To kill the rotating bit. If you miss the lit bit another one at that sense switch position will turn on, now leaving you two bits to destroy."

#815751
Author: John Trautschold
Length: 476 bytes
Multiplies a 12-digit floating point number times an 8-digit floating point number.

#88751
Author: Charlie Shields
Length: 22 bytes
Outputs 1's to an I/O port a selected percentage of the time. Outputs 0's the rest of the time.

#829751
Author: R. J. Walker
Length: 57 bytes
Loads checksummed 4K or 8K Altair BASIC from an unmodified PIO board.

#825751
Author: Robert L. Berg
Length: 34 bytes
Simple memory test. Halts when finds a bad location.

#818752
Author: Steve Phillips
Length: 477 bytes
Quadruple byte signed integer manipulation package. Addition, subtraction, multiplication, division as well as sign and zero value testing and other useful routines.

#813751
Author: John S. Robison
Length: 277 bytes
Tests ACR interface by comparing the record line output to the input from the monitor playback output from those tape machines which have monitor capability.

#94751
Author: Carl Swift
Length: 15 and 11 statements
BASIC programs to dump/store memory in octal using Altair BASIC's PEEK and POKE statements.

SUBROUTINES

#825754
Author: Dr. Jack W. Crenshaw
Length: 16 bytes
ASCII/HEX -- HEX/ASCII conversion subroutines.

#818751
Author: Jonathan Griffitts
Length: 23 bytes
Displays the contents of the accumulator in the address lights.

#811753
Author: Thomas D. Thomas
Length: 19 bytes
Adds a quadruple byte integer pointed to by [H,L] to registers [B,C,D,E].

#813752
Author: David Nowak
Length: 4 bytes
Subroutine to do a relative jump using a displacement in [D,E]. Could be used as an RST.

```
RJMP:  XTHL    ;Get return address in
               ;[H,L]
               ;Save [H,L] on stack
       DAD D   ;Add on displacement
       XTHL    ;Save back return
               ;address and restore
               ;[H,L]
       RET     ;Do relative branch
```

(Author's note: How about an RST that does a relative branch on the signed byte that follows the RST? Anyone?)

#84753
Author: Nilton G. Gimenes
Length: 33 bytes
Converts a 2-byte unsigned number into 6 individual octal digits in consecutive memory locations.

#811751
Author: Donald S. Tork
Length: 74 bytes
Searches a table of up to 255 entries of up to 255 byte strings for a match on the search string, which may also be up to 255 bytes long.

#99751
Author: Gary Tack
Length: 21 and 27 bytes
Routines to add/subtract two 16-digit BCD numbers.

#84752
Author: Nilton G. Gimenes
Length: 28 bytes
Translates six octal digits in consecutive locations in memory to a double byte value in [D,E].

#84751
Author: Nilton G. Gimenes
Length: 15 bytes for each routine
Octal/ASCII -- ASCII/octal conversion routines for octal digits/characters stored in six sequential memory locations.

NOTES ON PACKAGE I
By Paul Wasmund

One major point that was not covered in the Package I documentation was the stack. There are 30 bytes of stack available for the user at all times. If a program needs more stack space than that, it should set up a stack of its own.

If a program should infinite loop, stop your machine and restart the monitor by examining location Ø and pressing the run switch.

Release Two of Package I will be ready in early October. New features include tab stops to help make your program listings more readable and a highly improved editor. Two new commands have been added and all old editing commands have been improved.

FIND - The find command searches for a given string until a line containing it is found. The entire line it is in is then printed.

ALTER - Allows altering characters within a line. This command allows lines to be changed without having to replace them.

The insert and delete commands have been improved so that you can now insert blocks of lines and delete blocks of lines.

---continued on page 8

SOFTWARE

GENERAL SoftWARE UpDATE INFO

by Paul Allen

Here are a few items of interest:

1. The current version of BASIC being shipped is 3.1. The only change is that in the 8K version the array access is now significantly faster.

2. The FORTRAN cross assembler is now available on paper tape as well as a listing. The paper tape and listing together cost $30, while the listing alone costs $15. If you wish to obtain the cross assembler in another form, call the software department at the factory. The program order number is 521751.

3. There will be two versions of Extended BASIC; one that runs with the disk and one that runs without it. We are assuming that all orders placed are for the disk version so if you want the "stand-alone" version of Extended BASIC, drop us a line to let us know. The advantage of the stand-alone version is that it is 2K bytes smaller.

Programmed I/O

The coding technique for data input and output in which the CPU waits for completion of the I/O operation is usually termed "programmed I/O." This is by far the easiest and most common way of writing input and output subroutines for the Altair, and is used by BASIC and the Package I software.

There are usually two subroutines for each device. One that inputs a character from the device and one that outputs a character to the device. The input routine (INCHR) waits for the device's input buffer full flag to be set and then reads the character. On the Altair, the device status is in the input side of the lower I/O channel, and the data is read from that channel +1. Assuming we will return the byte read from the device in the A register, the code is as follows (for an old SIOC board--character ready bit in bit 5):

```
INCHR:  IN INCHN        ;where ICHN is the input channel
        ANI 40Q         ;TEST BIT 5=0 (Q means octal). The mask 40Q is
                        ;"anded" with the device status in the A register.
                        ;The mask (40Q) selects only bit 5.
        JZ INCHR        ;If no input data ready, loop.
        IN INCHN+1      ;Read the input byte.
        RET             ;Return from the subroutine.
```

Note that the input character routine is a "subroutine" that could be called many different places in a program by using a CALL instruction, i.e.

```
        .
        .
        .
        CALL INCHR      ;Get a character from the terminal.
        CPI 15Q         ;Was it a carriage return?
        JZ ENDLIN       ;If so, end of input line.
        .
        .
        .
```

Of course, the stack pointer must be set up pointing to an area of memory set aside for use by subroutine calls and PUSH/POP and other stack manipulations. This is most easily done as follows (this code is usually placed at the start of your program):

```
START:  LXI   SP, STACK
        .
        .
        .
        DS    20        ;set aside 20 locations (10 levels) of stack space
STACK:
```

More information on how to use the stack will be provided in Bill Gates' software article next month.

A corresponding character (byte) output subroutine for an old (REV 0) SIO board is listed below. The byte to be output is in the A register:

```
OUTCHR: PUSH PSW        ;Save the A register on the stack.
OUTLP:  IN INCHN        ;Read the device status into the A register.
        ANI 2Q          ;See if bit 1 is = 0.
        JZ OUTLP        ;If it is, keep waiting for the terminal to finish
                        ;printing.
        POP PSW         ;Get back the saved output byte.
        OUT INCHN + 1   ;Now output the byte to the terminal.
        RET             ;Return from subroutine.
```

> REMINDER:
> Users of BASIC and Package I should address their console I/O boards (SIOA, SIOC) for I/O port 0. The ACR board should be address for I/O port 6.

Often it is desirable to echo the character read from a terminal's keyboard immediately back to the terminal. The easiest way to do this is to insert

```
INECHO: CALL INCHR
```

right before the OUTCHR routine and then call INECHO instead of INCHR. If we knew we were always going to echo the input character back to the terminal, we could have the input character subroutine (INCHR) "fall into" the output character routine (OUTCHR). This may be done by placing INCHR directly ahead of OUTCHR and also removing the RET at the end of INCHR so an "OUTCHR" will always be performed when INCHR is called.

---continued on page 7

Slight modifications must be made to these routines if we want to use REV 1 or modified REV 0 serial I/O boards. In these boards, the character ready bit is in bit 0 of the status byte, and the character done (sent) bit is in bit 7. Also, the bits are "active low," that is, a 1 means the bit is false and a zero means the bit is true, which is just the opposite of the way the bits were set on the REV 0 board used in the previous examples. We could test bits by using an AND immediate instruction as before (i.e. replace the ANI 40Q in INCHR with an ANI 1Q and the ANI 2Q to an ANI 200Q) and changing the JZ's to JNZ's. However since the status bits are in the least and most significant bits in the status byte, we can conveniently test them by using the rotate instruction to move the bit in question into the carry flag and then using a JNC instruction to loop:

```
INCHR:  IN ICHN     ;Read status
        RAR         ;Character ready?
        JC  INCHR   ;If not, loop
        IN INCHN+1  ;Read character
        RET         ;Return

OUTCHR: PUSH PSW    ;Save character
OUTLP:  IN ICHN     ;Read status
        RAL         ;Test bit 7
        JC  OUTLP
        POP PSW     ;Get character
                    ;back in A
        OUT INCHN+1;Send it to
                    ;terminal
        RET         ;All done, return
```

Using rotates instead of ANIs saves one byte in each routine. Remember: taking care to save each byte you can will make long programs significantly shorter and faster.

PIO boards (often used for SWTPC TVTs) have the status bits "active low" like REV 1 SIO boards, but the status bits are in different positions: character ready is bit 1 and character done is bit 0, so:

```
INCHR:  IN   INCHN
        ANI  2Q
        JNZ  INCHR
        IN   INCHN+1
        RET

OUTCHR: PUSH PSW
OUTLP:  IN   ICHN
        RAR
        JNC  OUTLP
        POP  PSW
        OUT  INCHN+1
        RET
```

If you are confused by the use of "masks," here is an explanation. If we want to make a jump on only one bit of the A register, we "and" a mask with that bit on with A. The result of the AND will be zero if that bit was zero, and non-zero if the bit was one. Here is a table of bit masks (in octal) for each bit position:

BIT	MASK	
0	1	(usually use RAR to test)
1	2	
2	4	
3	10	
4	20	
5	40	
6	100	
7	200	(usually use RAL to test)

Note that bits 0 and 7 take fewer bytes to test than the rest because they can be rotated into the carry status bit as mentioned earlier.

It is often very useful to use bit testing and setting in a program. Suppose you are writing an assembler and you want to remember if you have seen any colons or commas on a line. You could use one bit in a register to flag the fact you had seen a colon and another bit to flag whether you had seen a comma; and you could use the other six bits of the register for six other flags. Suppose the flags were kept in the B register. Then, to set a flag (if bit=1 means set):

```
        .
        .
        .
   MOV A,B    ;Get flag register in A

   DRI 2      ;Mark colon seen (bit 1)

   MOV B,A    ;Save flags back
```

To reset a flag:

```
   MOV A,B    ;Get flag register in A

   ANI 375    ;377-2
              ;Reset colon flag (bit 1)

   MOV B,A    ;Save flags back
```

To test two flags:

```
   MOV A,B    ;Get flag register

   ANI 12Q    ;Test both bits 3 & 1
              ;(colon and comma)

   JZ NETHER  ;Jump to NETHER if both
              ;flags = 0

   JNZ ONEFLG ;Jump to ONEFLG if one
              ;or both of two flags
              ;set.
```

To complement (invert) a flag (reset it if set, set it if reset):

```
   MOV A,B    ;Get flag register

   XRI 2      ;Flip (complement) bit 1

   MOV B,A    ;Save flags back
```

Final Note:

If you'd like us to cover some particular technique or coding practice in detail, let us know.

Software Notes

by Bill Gates

Though the most difficult and enjoyable part of writing a program is the design of data structures and program flow, it is also important to use the least number of instructions possible to perform each function in a program. For instance:

```
CALL SUB1       should be replaced by
RET

JMP SUB1        unless something fairly
                tricky is being done
```

with return addresses. The JMP is faster, takes one less byte, and uses no stack space. An instruction book on programming the 8008 ignores this simple fact!

JMPs should be avoided wherever possible. By rearranging code you can often avoid having an unconditional JMP by falling into the routine you were JMPing to.

The beginning programmer will use lots of SHLDs, LHLDs, STAs and LDAs when they are not necessary. The stack can be used to save temporary values in most cases. SHLDs, LHLDs, LDAs and STAs should only be used for values referenced in many different contexts within a program, i.e. an I/O parameter or the current line number.

A good technique for familiarizing yourself with the instruction set is to go out of your way to use every instruction at least once (except perhaps DAA). Go through the instruction set from time to time and look closely at the instructions you seem to use very rarely. With few exceptions (DAA, SPHL) all the instructions can be used to advantage, even in small programs. One of the most overlooked instructions is XTHL. When all the accumulators have values that must be saved and a value needs to be taken off the stack, XTHL is the only instruction that can be used.

```
Example:  ;Exchange [B,C] with [H,L]

PUSH B    ;put [B,C] on the stack

XTHL      ;[H,L] = top stack entry =
           [B,C]

          ;[H,L] goes on the stack

POP B     ;[B,C] = original [H,L]
```

Sometimes the simple way of doing things is the best. PUSH B/POP D may seem like a tricky way of setting [D,E] = [B,C], but the obvious sequence MOV D,B/MOV E,C is much faster.

---continued on page 8

Some tricks involve instruction sequences which at first sight seem meaningless. For instance: SUB A or XRA A. Subtracting A from itself or exclusive-oring A with itself are the only one-byte ways of setting A=0. MVI A,0 must still be used if the condition codes need to be preserved, but this is rare.

ADC A is equivalent to RAL, except it affects all the condition codes. SBB A sets A=0 if carry is off and A=377 if carry is on. The routine below uses this fact to convert A as a signed integer to a double byte signed integer in [H,L]:

```
MOV L,A  ;setup the low order
         ;now the sign must be
         ;"extended" by setting H=0
         ;if A=>0 and H=377 otherwise
RAL      ;Carry = 1 if A<0
         ;Carry = 0 if A=>0
SBB A    ;A=0 if old A was =>0
         ;A=377 if old A was <0
MOV H,A  ;setup the high order
```

The sequence: INR E
DCR E
doesn't modify any values, but it does set the condition codes (except carry) depending on what is in E. If E is being used as a flag to indicate, say, whether or not a decimal point has been seen, the zero flag is set up to do a conditional JMP.

The subject of good decimal print routines has been discussed extensively in the Altair Software Department this week. This routine is one of the four or five I wrote this week -- each with its own advantages and disadvantages. This one is fairly tricky, in that it takes a little bit of looking at to understand.

```
#1 ;
   ;Print the binary unsigned number
   ;in [H,L] in decimal, suppressing
   ;leading zeros
   ;
   ;24 bytes (25 if saves D,E)
   ;ON RETURN:
   ;A = last digit in ASCII
   ;B,D = 255 (all constants in
   ;decimal)
   ;C,E = last digit -1Ø
   ;H,L = Ø
   ;
   ;Uses up to 18 bytes of stack
   ;Total compute time up to 85
   ;milliseconds
   ;
   ;IDEA: calculate a digit, save it
   ;      on the stack, and call the
   ;      digit calculator to calcu-
   ;      late and print higher order
   ;      digits, pop the digit off
   ;      and print it.
   ;
DECOUT: LXI B, -1Ø       ;CALL here
GETDIG: MOV D,B          ;[D,E] = -1
        MOV E,B          ;since B = 255
LOOPSB: DAD B            ;Subtract 1Ø from [H,L] until [H,L] < 1Ø. Carry
                         ;won't be set by the last DAD when [H,L] < 1Ø.
        INX D            ;increment the count
        JC LOOPSB        ;loop subtracting
        PUSH H           ;[L] = current digit -1Ø
                         ;Save the current digit on the stack.  Change to
                         ;XTHL and add PUSH D at GETDIG to save [D,E].
        XCHG             ;[H,L] = old [H,L]/1Ø
        MOV A,H          ;Set zero flag if [H,L] = Ø
        ORA L
        CNZ GETDIG       ;If not zero, print the higher order digits and
                         ;then return here to print this digit.
        MVI A, "Ø" + 1Ø  ;A = constant to add to digit
        POP B            ;pop the digit into C
        ADD C            ;A = ASCII of digit
        JMP OUTCHR       ;Jump to the routine to print A and return.  If
                         ;OUTCHR is located next, the JMP can be eliminated.
```

Parity is used as a check to detect errors in data transmission. Each data word is given an additional bit which is set to 1 if there are an odd number of 1's in the data and Ø otherwise. When the data is received the parity bit is checked to make sure it is set properly. Thus, if you are reading a 7-bit ASCII paper tape with the 8th bit used for parity, the parity of the entire 8 bits should be even.

The reason I first thought about a parity routine for the 8080 is that the parity condition code and all the instructions related to it (JPO, JPE, RPE, RPO, CPO, CPE) are seldom used. I wondered how difficult it would be to calculate parity if the parity flag were removed. A user-settable flag would be much more useful than the parity flag. BASIC uses the parity flag in only about eight places, and all of these are special tricks. Here is the smallest parity routine I've been able to write:

```
;Enter with number in A.  1Ø bytes.
;On exit, A=Ø and all the other reg-
;isters are preserved.
;Carry is set depending on A's
;parity.
;Enter at ODDPAR for carry on to
;mean odd parity.
ODDPAR: ADD A            ;Move a bit of A into carry.
        RZ               ;If all bits added into carry, return.
        JNC ODDPAR       ;If no bit moved into carry, rotate more.
;enter at EVNPAR for carry on to
;mean even parity
EVNPAR: ADI 200          ;Complement the parity of the remaining bits
        JMP ODDPAR       ;Rotate more.
```

I said last month I would explain the bootstrap loader but I've decided that should wait until next month when I explain the basics of the stack.

Also next month: multiprecision arithmetic, and more interesting subroutines.

WHICH I/O INTERFACE FOR YOU?

1. SIOC-
   For Teletypes* or other 20mA current loop asynchronous terminals up to 19,200 baud. (5-8 data bits)

2. SIOA-
   For asynchronous RS-232 CRTs or other terminals of data rates up to 19,200 baud. (5-8 data bits)

3. SIOB-
   Same as SIOA and SIOC except output and input are TTL compatible levels.

4. PIO-
   For bidirectional transmission of bytes at speeds up to approximately 25,000 bytes/sec (200,000 baud). Eight lines (1 byte) in and out plus "handshaking." All lines standard TTL compatible. Most commonly used for SWTPC-TVTs or equivalent, custom A/D-D/A interfaces, computer to computer interfaces, numerical control applications.

*Teletype is a registered trademark of the Teletype Corporation.

Package 1 continued--

Another large improvement was made in line number specifications. In addition to being able to say Print line 5 (P5) you now can also say print the current line (P.) or print the current line plus or minus a constant (P.-6, .+6).

Also, typing escape will print the previous line, and line feed will print the line after the current one.